Amendments to the Claims:

The listing of Claims will replace all prior versions and listings of the Claims in the application:

Listing of Claims

1. – 20. (Canceled)

21. (Currently Amended) A method of operating a business services application for retrieving data with delivery technologies, the method comprising:

receiving at a server a request for information initiated with a delivery technology;

~~developing custom application code in a subclass of a BusinessService class, the custom application code responsive to a request for data initiated by the delivery technologies;~~

translating the request to a first document object model document with an ApiService class operable on the server;

during the translation with the ApiService class to the first document object model document, limiting with a Message class that is a wrapper of the document object model document a first ~~the~~ data structure of the first document object model document to representation as an input message with a plurality of fields, wherein the first data structure ~~units of data included in each of the fields~~ is limited by a plurality of methods included in the Message class to represent a field data type that is pre-specified with the methods ~~in the business services application;~~

executing ~~the~~ custom application code to retrieve data based on the first document object model document;

reading said data into a second document object model document with the custom application code ~~ApiService class~~;

while said ~~the~~ data is being read in with the custom application code, limiting a second ~~the~~ data structure of the second document object model document with the methods included in the Message class to representation as an output message with a plurality of fields, wherein the second data structure ~~units of data included in each of the fields~~ is limited with the methods to represent the field [[a]] data type that is pre-specified with the methods ~~in the business services application~~; and

translating the second document object model document with the ApiService class to generate an output message in a format that is compatible with ~~based on~~ the delivery technology.

22. (Currently Amended) The method of claim 21, wherein limiting the first data structure of the first document object model document comprises populating a plurality of text nodes within the first document object model document with request parameters contained in the request that are translated with the methods to a format identified with the field ~~pre-specified~~ data type.

23. (Canceled)

24. (Currently Amended) The method of claim 22, wherein limiting the first data structure of the first document object model document further comprises limiting the field ~~predetermined~~ datatype to ~~a format of~~ a string datatype.

25. (Currently Amended) The method of claim 21, wherein limiting the first data structure of the first document object model document comprises populating an attribute node within the first document object model document with an attribute of the request that is translated to a format identified with the field ~~pre-specified~~ data type.

26. (Previously Presented) The method of claim 21, further comprising selecting, as a function of a mode debug flag, to use one of a short field name or a long field name as a field name for each of the fields in the first and second document object model documents.

27. (Currently Amended) The method of claim 21, wherein the field ~~pre-specified~~ data types is ~~are~~ selected from a pre-specified group of data types consisting of a string datatype, a long datatype, an integer data type, a boolean data type and a group data type.

28. (Currently Amended) The method of claim 21, wherein limiting the first data structure of the first document object model document comprises loading a static declaration of a datatype based on a list of fields expected in the request.

29. (Currently Amended) The method of claim 21, wherein limiting the second data structure of the second document object model document comprises populating a plurality of text nodes within the second document object model document with said data read in to the second document

object model document, wherein the format of said ~~the~~ data that is read in is converted based on the field data type.

30. (Currently Amended) The method of claim 21, wherein limiting the second data structure of the second document object model document comprises populating an attribute node within the second document object model document with an attribute read in to the second document object model document that is translated to a format identified with the field ~~pre-specified~~ data type.

31. (Canceled)

32. (Previously Presented) The method of claim 21, wherein translating the second document object model comprises translating the second document object model document to extensible markup language text.

33. (Previously Presented) The method of claim 21, wherein translating the second document object model comprises translating the second document object model document to at least one of a hypertext markup language and a website meta language as a function of at least one extensible stylesheet language stylesheet.

34. – 40. (Canceled)

41. (Currently Amended) A system ~~for leveraging extensible markup language technology~~ to provide an interface between a back-end systems layer and a front-end systems layer, the system comprising:

a server computer;

an ApiService class operable within the server computer to translate ~~direct the translation of~~ a request to an input message that includes a plurality of fields;

a document object model class operable within the server computer to represent the input message as a document object model document;

a Message class ~~and a Field class~~ operable within the server computer as a wrapper of the document object model class to restrict ~~manipulation and standardize the content~~ a format of the document object model document;

a MESSAGEDEFINITION class operable in the server, wherein the MESSAGEDEFINITION class includes a listing of pre-specified fields each of which describe a corresponding pre-specified data type, and wherein the Message class is ~~and the Field class are~~ further operable, within the server during translation, to limit a format of ~~corresponding~~ fields included in the input message to a predetermined data structure based on the described corresponding pre-specified data type; ~~and~~

a BusinessService class operable within the server computer to execute ~~direct the execution of~~ custom application code ~~as a function of~~ based on the input message, wherein the custom application code includes a pre-specified data type to limit the format of those fields included in

the input message that do not correspond to the listing of pre-specified fields, the custom application code is operable to process the input message to retrieve data,

the custom application code and the Message class operable to translate the data to an output message that is transmitted by the server in response to the request.

42. (Currently Amended) The system of claim 41, wherein ~~the custom application code is operable to process the input message to retrieve data, the data translatable with the document object model class, the Message class and the Field class to an~~ the output message is in the form of a document object model document with restricted ~~manipulation~~ and standardized content based on the pre-specified data type included in the custom application code that, during translation with the Message class, is operable to limit a format of each of a plurality of fields included in the output message to a predetermined data structure.

43. (Currently Amended) The system of claim 41 [[2]], wherein the ApiService class is operable to direct the conversion of the output message to a presentation format defined by the request.

44. (Previously Presented) The system of claim 41, wherein the input message and the output message comprises a root element and a plurality of sub-elements.

45. (Previously Presented) The system of claim 41, further comprising a Fldtypes class operable within the server computer, wherein the Fldtypes class comprises definitions of the format of datatypes for fields within the input message.

46. (Previously Presented) The system of claim 41, wherein the document object model document comprises a plurality of field names, the field names selectable with a mode debug flag as one of a short field name and a long field name.

47. (Previously Presented) The system of claim 46, wherein the short field name and the long field name are defined in the MESSAGEDEFINITION class operable within the server computer.

48. (Previously Presented) The system of claim 41, wherein the document object model class comprises a Document class, a document object model Element class and a plurality of ProcessingInstruction classes, the Message class operable as a wrapper of the Document class, the document object model Element class and the Processing Instruction classes.

49. (Currently Amended) The system of claim 41, wherein the document object model class comprises a document object model setAttribute method, and a Field class operable as a wrapper of the document object model setAttribute method.

50. (Currently Amended) The system of claim 41, wherein the BusinessService class comprises a subclass of the custom application code responsive to the request.

51. – 63. (Canceled)

64. (Currently Amended) An e-commerce architecture for providing a framework to interface delivery technologies with data, the e-commerce architecture comprising:

a server computer operable to execute instructions to convert a request from a delivery technology to a first document object model document in an extensible markup language, the first document object model document comprising a plurality of request parameters extracted from the request;

the server computer operable to execute instructions included in a Message class that is operable as a wrapper of the document object model document to restrict the conversion to the first document object model document based on a listing of data types that are pre-specified for the request parameters by a plurality of methods included in the Message class, wherein the data types limit a first ~~the~~ data structure of a plurality of fields included in the first document object model document to a predetermined data structure specified by the data types;

the server computer operable to execute instructions to retrieve data that may be responsive to the request, and convert the data to a second document object model document in the extensible markup language, based on the request parameters; ~~and~~

the server computer operable to execute instructions included in a Message class that is operable as a wrapper of the document object model document to restrict the conversion of the data to the second document object model document to limit a second ~~the~~ data structure of a plurality of fields included in the second document object model document to the [[a]] predetermined data structure specified by the methods included in the Message class ~~data types;~~ and

the server computer operable to execute instructions to generate an output message in a format that is compatible with the delivery technology.

65. (Previously Presented) The e-commerce architecture of claim 64, wherein the instructions to restrict the conversion of the first and second document object model documents further comprise instructions executable by the server computer to identify the first and second document object model documents with a predefined name included in the request.

66. (Previously Presented) The e-commerce architecture of claim 64, wherein the instructions to restrict the conversion of the first and second document object model documents further comprise instructions executable by the server computer to create a plurality of element nodes and populate a plurality of corresponding text nodes with the respective request parameters and the respective data.

67. (Previously Presented) The e-commerce architecture of claim 66, wherein the instructions to restrict the conversion of the first and second document object model documents further comprise instructions executable by the server computer to define the data type of each of the text nodes from among a predefined group of data types.

68. (Currently Amended) The e-commerce architecture of claim 64, wherein the ~~instructions to restrict the conversion comprises a~~ Message class is operable as a wrapper of a plurality of classes included within a ~~the~~ document object model class that include a document class and a document object model element class.

69. (Previously Presented) The e-commerce architecture of claim 64, wherein the instructions to restrict the conversion comprises a Field class operable as a wrapper of a document object model setAttribute method in a document object model element class.

70. (Previously Presented) The e-commerce architecture of claim 64, wherein the instructions to retrieve data responsive to the request are identified with a request name that is included in the request.

71. (Currently Amended) The method of claim 21, wherein limiting the first data structure of the first document object model comprises standardizing the format of the document object model

to be substantially similar for a similar request received from any one of a plurality of different ~~the~~ delivery technologies.

72. (Currently Amended) The method of claim 71, wherein limiting the second data structure of the second document object model comprises standardizing the format of the second document object model to be compatible with any one of the different delivery technologies.

73. (Currently Amended) The method of claim 72, wherein executing the custom application code comprises executing the same custom application code for a similar request from any one of the different delivery technologies to provide a response.

74. (Currently Amended) The method of claim 21, wherein executing the custom application code comprises executing the same custom application code for a similar request from any one of a plurality of different ~~the~~ delivery technologies.

75. (Currently Amended) The method of claim 74, wherein while the data is read in, limiting the second data structure of the second document object model document comprises similarly limiting the second document object model in response to similar requests from any of the different delivery technologies.

76. (Currently Amended) The system of claim 41, wherein the Message class ~~and the Field class are~~ is operable, during representation of the input message as the document object model document, to restrict manipulation of the document object model document.

77. – 78. (Canceled)

79. (Currently Amended) The method of claim 41, wherein the pre-specified datatypes is ~~are~~ selected from the group consisting of integer, long, Boolean, string and group.

80. (New) The method of claim 21, wherein translating the request comprises translating the request to a first instance of the Message class with the ApiService class, and creating the first document object model document with the first instance of the Message class.

81. (New) The method of claim 21, wherein the Message class comprises a plurality of instances of a Field class, the Field class operable as a second wrapper of a document object model Element class associated with the first document object model document.

82. (New) The method of claim 81, wherein translating the request comprises translating the request to a first instance of the Message class with the ApiService class by adding an instance of the Field class to the first instance of the Message class for each of a plurality of request

parameters included in the request, and creating the first document object model Element class with the instance of the Field class for each of the request parameters.

83. (New) The method of claim 80, wherein translating the request to the first instance of the Message class comprises adding an instance of a Field class to the first instance of the Message class for each of a plurality of request parameters included in the request.

84. (New) The method of claim 80, wherein executing custom application code to retrieve data comprises executing the custom application code with the ApiService class to retrieve said data based on the first instance of the Message class.

85. (New) The method of claim 84, wherein executing custom application code further comprises the custom application code storing said data in a second instance of the Message class through use of at least one of the methods included in the Message class, and returning to the ApiService class the second instance of the Message class.

86. (New) The method of claim 85, wherein reading said data into a second document object model document comprises reading the document object model document from the second instance of the Message class with the ApiService class.

87. (New) The method of claim 21, wherein translating the second document object model document with the ApiService class comprises translating the second document object model into one of XML or a presentation format based on the request.

88. (New) The method of claim 87, wherein the presentation format is HTML.

89. (New) The method of claim 87, wherein translating the second document object model further comprises using at least one XSL stylesheet to transform the second document object model to the presentation format.

90. (New) The method of claim 85, wherein storing said data in a second instance of the Message class further comprises:

translating a value of said data, with at least one of the methods of the Message class, to a text value; and

storing the text value in the second document object model document, the text value is one of a group of data types comprising a long integer data type, an integer data type, or a boolean data type.

91. (New) The method of claim 85, wherein storing said data in a second instance of the Message class further comprises storing data type information from a MESSAGEDEFINTION

class and a value in the second document object model document, the value generated with at least one of the methods of the Message class.

92. (New) The system of claim 41, wherein the Message class is a first wrapper of the document object model document, and the system further comprises a Field class that is a second wrapper of a document object model Element class included in the document object model class, the Field class is operable to restrict a value stored in an instance of the document object model Element class.

93. (New) The system of claim 41, wherein the Message class comprises a plurality of methods, and a format of a data structure of the document object model document is limited to data types that are pre-specified with the methods.

94. (New) The system of claim 41, wherein the ApiService class is further operable to translate the output message to an XML format, wherein the ApiService class is operable to transmit the XML format as a response to the request.

95. (New) The system of claim 92, wherein the Message class is operable to restrict creation of element nodes included in the document object model document and population of text nodes that correspond to the element nodes, and the Field class is operable to restrict creation of attribute nodes included in the document object model document.

96. (New) The system of claim 41, wherein the ApiService class is further operable to translate the output message to a presentation format using a plurality of XSL stylesheets and to transmit the presentation format as a response to the request.

97. (New) The system of claim 96, wherein the ApiService class is further operable to select the plurality of XSL stylesheets based on a format of the request.